DOCUMENT RESUME

ED 108 608 IR 002 119

AUTHOR Kline, Lanaii
TITLE Story-List.
INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.
REPORT NO SWRL-TN-5-72-01
PUB DATE 21 Jan 72
NOTE 19p.; Several pages may reproduce poorly due to light print

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE
DESCRIPTORS *Computer Programs; Computer Storage Devices; Educational Development; Elementary Education; Flow Charts; *Information Storage; Input Output; Reading Materials; *Vocabulary; *Word Lists
IDENTIFIERS *STORY LIST

ABSTRACT

The program, STORY-LIST, generates alphabetized cumulative word lists by story number within a school grade. It is designed to read a group of cards until it finds a new grade/story number. Each word read is stored in an array, sorted, and an asterisk is added to each word in the array. This array is then merged with the old sorted word list and printed. The new array formed by the merge is stored in the old word list array after deleting the asterisks from the new words. This process continues until the input data is exhausted. Included in this document are a flow chart layout form, sample data, and sample run. (Author/SK)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED108603

SWRL

SOUTHWEST REGIONAL LABORATORY TECHNICAL NOTE

DATE: January 21, 1972

NO: TN 5-72-01

SCOPE OF INTEREST NOTICE
The ERIC Facility has assigned this document for processing to: IR | CS
In our judgement, this document is also of interest to the clearinghouses noted to the right. Indexing should reflect their special points of view.

TITLE: STORY-LIST

AUTHOR: Lanaii Kline

ABSTRACT

This document describes the program, STORY-LIST which prepares an alphabetized cumulative vocabulary list for each story within a school grade.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 11300 La Cienega Boulevard, Inglewood, California, 90304.

IR 000 119

1.0 - PROGRAM IDENTIFICATION

STORY-LIST

2.0 - OBJECTIVE

STORY-LIST generates alphabetized cumulative word lists by story number for a grade. Each word which is introduced to that grade and story number is printed with an asterisk.

3.0 - PROGRAM DESCRIPTION

3.1 - Program Logic

The program reads a group of cards until it finds a new grade/story number. Each word read is stored in an array, sorted, and an asterisk is added to each word in the array. This array is merged with the old sorted word list and is printed. The array formed by the merge is stored in the old word list array after deleting the asterisks from the new words. This process continues until the input data is exhausted.

3.2 - Variables and Internal Tables

Character Arrays:

- ARRAY(2000) - is an array of character strings of length 20 resulting from the merge of the arrays DATA and OLD.
- ASTER(2000) - is an array of character strings of length one. The value of each element is an asterisk if the corresponding word in ARRAY originated from DATA or a blank if the corresponding word in ARRAY originated from OLD.
- DATA(200) - is an array containing the new input words. This array is sorted when a new story group is detected. The elements of this array are of length 20.
- OLD(2000) - is an array whose elements are of length 20 and whose values are the result of merging the new data, DATA, with the old data, OLD, and assigning the merged data to OLD.

Character Variables:

- IDUMMY - is a dummy variable of length 20 which is used when the array DATA is being sorted.
- DUMMY - is a dummy variable of length one which is used to pad the input statement.

GRADE - is an input variable of length five which contains the grade and story number of the input word.

WORD - is the input word of character length 20.

OLD_GRADE - is a variable of length five which contains the current value for the grade/story number.

STORY - is a variable of length four which is derived from the last four characters in OLD_GRADE.

YR - is a variable of length 12 varying whose value is derived from the first character in OLD_GRADE. If SUBSTR(OLD_GRADE,1,1)='K', then YR='KINDERGARTEN'.

Integer Variables:

I - is the number of entries in the array DATA.

IFLAG - is a flag which indicates that the first story group is being read (IFLAG=1), that the first story group is to be printed and not merged (IFLAG=2), or that the rest of the words are to be handled normally (IFLAG=0).

J - is an index for a DO-loop.

M - is an index for a DO-loop.

N - is the number of entries in the arrays, ARRAY and ASTER.

NEND - is a flag which indicates that the last group of words has been read.

NFLAG - is a flag which indicates that the order of the array DATA has been changed and to continue the sort.

NOLD - is an index for the array OLD.

NUMB - is the number of entries in the array OLD.

QFLAG - is a flag which is added to the program if the production run terminates for not enough time, pages, or I/O requests. See Attachment 5.

3.3 - Input/Output Arguments
Not Applicable

4.0 - SUBROUTINES AND FUNCTIONS

Not Applicable

5.0 - DATA SPECIFICATIONS

5.1 - Input Formats
See Attachment 1.

5.2 - Output Formats
See Attachment 2.

6.0 - PROGRAM CONSTRAINTS AND LIMITATIONS

6.1 - Programming Language
PL/1

6.2 - Vendor
Campus Computing Network, University of California, Los Angeles, California

6.3 - Storage Requirements
Compilation: 162K
Execution: 140K

6.4 - Hardware Configuration
IBM 360/91, punched cards

6.5 - Program Parameters
Currently the maximum number of words which can be processed is 2000 and the maximum number of input words for any one story group is 200. These numbers can be altered by changing the appropriate cards in the program deck. The maximum length a word may have is twenty characters.

6.6 - Error Messages
If the DATA array exceeds 200, the program terminates after printing a message saying that the input for the current story group exceeded 200. If the array ARRAY exceeds 2000, the program terminates after printing a message saying that the word list exceeded 2000. See Attachment 4.

7.0 - OPERATING INSTRUCTIONS

The program is submitted in a batch mode with the following cards:

```
col 1
//AGB005XX JOB 't,i/o,R,ID=SWRL,BIN=0738','name'
/, PASSWORD abcabcac
// EXEC PL1LFCG
//PL1.SYSIN DD *

    program deck

/*
```

```
//GO.SYSIN DD *
    data deck
/*
//
```

See Attachment 3 for Sample Data and Attachment 4 for Sample Run.

8.0 - PROGRAM FLOWCHART

Attachment 2

SWRL

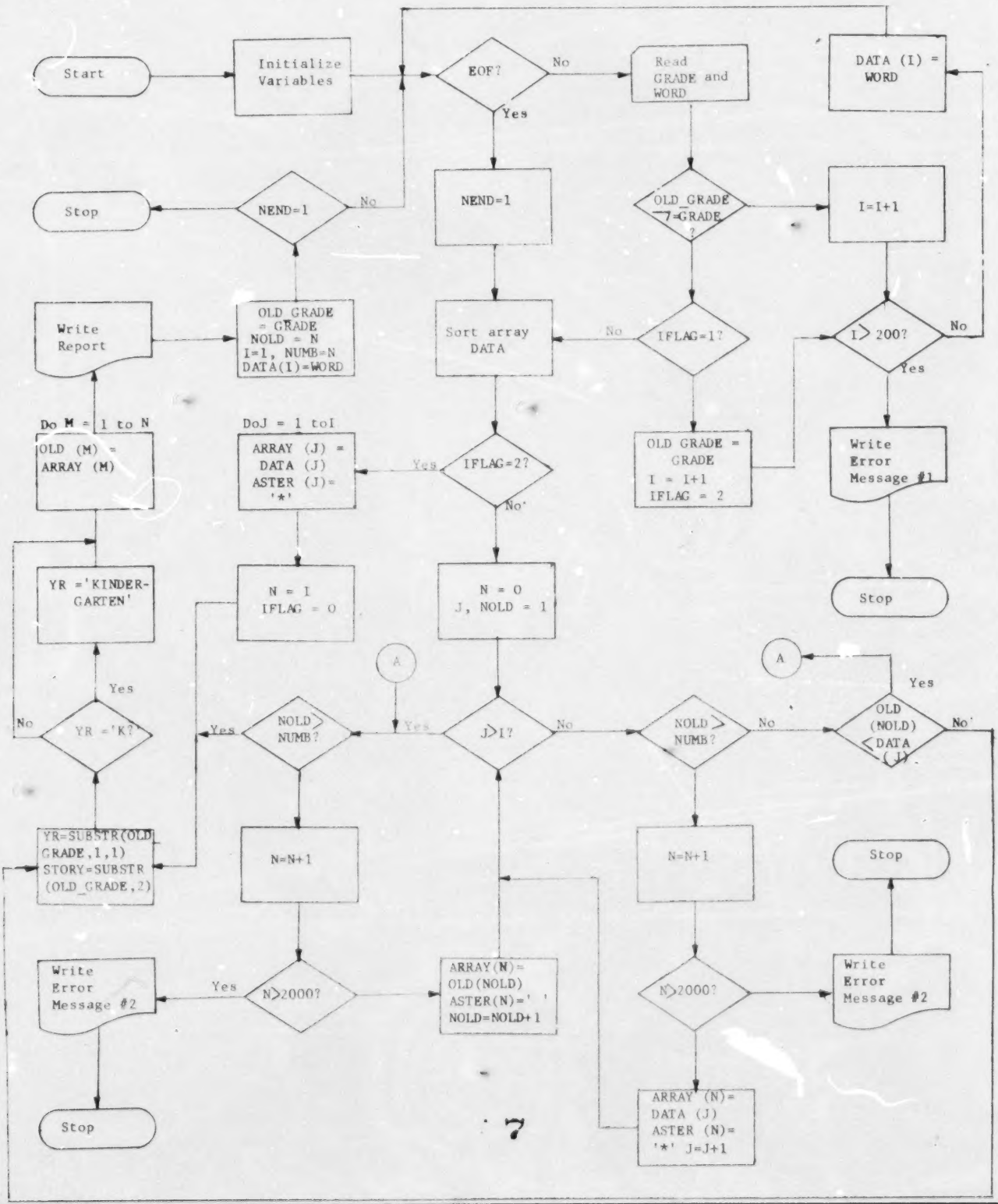
FLOWCHART LAYOUT FORM

PROGRAMMER: Lanaii Kline

PROGRAM ID: STORY-LIST

DATE: May 4, 1972

PAGE 1 OF 1



9.

9.0.- PROGRAM LISTING

Attachment 1

```
PROG:PROC OPTIONS (MAIN);
     DCL OLD_GRADE CHAR(5);
     DCL YR CHAR(12) VARYING,STORY CHAR(4);
     DCL DATA(200) CHAR(20),WORD CHAR(20),GRADE CHAR(5);
     DCL ARRAY (2000) CHAR(20);
     DCL ASTER(2000) CHAR(1);
     DCL DUMMY CHAR(1),IDUMMY CHAR(20);
     DCL OLD(2000) CHAR(20);
     OLD_GRADE='     ';
     IFLAG=1;
     N,I,NEND,NOLD=0;
     ON ENDFILE(SYSIN) GO TO DONE;
READ:GET EDIT(GRADE,WORD,DUMMY) (A(5),X(4),A(20),X(50),A(1));
     IF OLD_GRADE ^= GRADE THEN GO TO A;
                           ELSE DO;
                                I=I+1;
                                IF I>200 THEN GO TO DATAFLOW;
                          CONCAT:DATA(I)=WORD;
                                GO TO READ;
                                END;
A:IF IFLAG=1 THEN DO;
                  OLD_GRADE=GRADE;
                  I=I+1;
                  IF I>200 THEN GO TO DATAFLOW;
                  IFLAG=2;
                  GO TO CONCAT;
                  END;
             ELSE LAST:DO;
  /* SORT NEW WORDS */
                C:NFLAG=0;
             LOOP:DO J=1 TO I-1;
                  IF DATA(J) > DATA(J+1) THEN DO;
  /* CHANGE ORDER OF VARIABLES */
                                              IDUMMY=DATA(J);
                                              DATA(J)=DATA(J+1);
                                              DATA(J+1)=IDUMMY;
                                              NFLAG=1;
                                              END;
                  END LOOP;
                  IF NFLAG=1 THEN GO TO C;
                  IF IFLAG=2 THEN DO;
                                  DO J=1 TO I;
                                  ARRAY(J)=DATA(J);
                                  ASTER(J)='*';
                                  END;
                                  N=I;
                                  IFLAG=0;
                                  GO TO B;
                                  END;
  /* MERGE TWO DATASETS */
                  N=0;
                  J=1;
                  NOLD=1;
             REED:IF J>I THEN GO TO MVOLD;
                  IF NOLD>NUMB THEN GO TO MVDATA;
                  IF OLD(NOLD)<DATA(J) THEN MVOLD:DO;
```

```
                                        IF NOLD>NUMB THEN GO
                                                    TO B;
                                        N=N+1;
                                IF N>2000 THEN GO TO ARRFLOW;
                                        ARRAY(N)=OLD(NOLD);
                                        ASTER(N)=' ';
                                        NOLD=NOLD+1;
                                        GO TO REED;
                                        END;
                                ELSE MVDATA:DO;
                                        N=N+1;
                                IF N>2000 THEN GO TO ARRFLOW;
                                        ARRAY(N)=DATA(J);
                                        ASTER(N)='*';
                                        J=J+1;
                                        GO TO REED;
                                        END;

    /* WRITE OUTPUT */
                        B:YR=SUBSTR(OLD_GRADE,1,1);
                          STORY=SUBSTR(OLD_GRADE,2);
                          PUT PAGE;
                          IF YR='K' THEN YR='KINDERGARTEN';
                          DO M=1 TO N;
                          OLD(M)=ARRAY(M);
                          END;
                          PUT SKIP EDIT('GRADE',YR,'STORY NUMBER',STORY)(X(60),
                                    A(5),SKIP(2),X(58),A(12),X(1),SKIP(4),
                                    X(58),A(12),SKIP(2),X(60),A(4),
                                    SKIP(2));
                          PUT SKIP;
                          PUT EDIT ((ASTER(M),ARRAY(M) DO M=1 TO N))(4(A(1),
                                    X(1),A(20),X(5)),SKIP);
                          OLD_GRADE=GRADE;
                          NUMB=N;
                          NOLD=0;
                          I=1;
                          DATA(I)=WORD;
                          IF NEND=1 THEN GO TO FINISH;
                          GO TO READ;
                          END;
DONE:NEND=1;
     GO TO LAST;
DATAFLOW:PUT EDIT(' THE NUMBER OF WORDS IN THIS STORY GROUP',OLD_GRADE,
                  'EXCEEDED 200 WORDS')(SKIP(2),A(40),X(1),A(5),A(18));
         GO TO FINISH;
ARRFLOW:PUT EDIT(' THE NUMBER OF WORDS EXCEEDS 2000 AT STORY GROUP',
                  OLD_GRADE)(SKIP(2),A(48),X(1),A(5));
FINISH:END PROG;
```

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

TITLE: _____ DATE: _____

FILE ID: Card Input

PROGRAMMER: Lanaii Kline

RECORDING MODE:
FIXED [X]
LENGTH 80
VARIABLE [ ]
MAX _____
MIN _____

BLOCKING FACTOR:
BLOCK CONTAINS
[ ] RECORDS

REMARKS:

CARD LAYOUT [X]
TAPE RECORD LAYOUT [ ]

FORMAT LEGEND:

△ = Blank	P = Packed Dec.
A = Alpha	O = Octal
N = Numeric	H = Hexadecimal
X = Alpha/Num.	B = Binary

RECORD I.D.

	STORY NUMBER		WORD	
FORMAT	A	△	A	△
POSITIONS	5	10	15	20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED
RECORD I.D.

FORMAT
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED
RECORD I.D.

FORMAT
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED
RECORD I.D.

FORMAT
POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

10

SWRI

COMPUTER SYSTEMS
FILE/RECORD LAYOUT

RECORDING MODE:
FIXED [X]
LENGTH 132
VARIABLE [ ]
MAX
MIN

REMARKS:

TITLE: DATE:

FILE ID: Output

PROGRAMMER: Lanaii Kline

BLOCKING FACTOR:
BLOCK CONTAINS
[ ] RECORDS

PRINT FILE LAYOUT [X]

TAPE RECORD LAYOUT [ ]

FORMAT LEGEND:
△ = Blank P = Packed Dec.
A = Alpha O = Octal
N = Numeric H = Hexadecimal
X = Alpha/Num. B = Binary

RECORD I.D.

* or blank	Word	* or blank	Word	* or blank	Word	* or blank	Word
A	A	A	A	A	A	A	A

FORMAT

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[X] CONTINUED

RECORD I.D.

word (cont'd)		
A	A	

FORMAT

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED

RECORD I.D.

FORMAT

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

[ ] CONTINUED

RECORD I.D.

FORMAT

POSITIONS 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100

K001	I
K001	WILL
K001	GO
K002	PLAY
K002	THERE
K003	WITH

ATTACHMENT 4
Sample Run

GRADE

KINDERGARTEN

STORY NUMBER

C01

* GO * I * WILL

657576

GRADE

KINDERGARTEN

STORY NUMBER

002

GO
WILL

I

* PLAY

* THERE

657575

GRADE

KINDERGARTEN

STORY NUMBER

003

GO	I	* ME	PLAY
THERE	WILL	* WITH	

657574

GRADE

KINDERGARTEN

STORY NUMBER

008

ANN	ARE	GO	I
* IN	IT	LOG	ME
* NAT	ON	PLAY	THE
THERE	TO	WE	WILL
WITH	YOU		

THE NUMBER OF WORDS EXCEEDS 2000 AT STORY GROUPK009

GRADE

KINDERGARTEN

STORY NUMBER

008

ANN	ARE	GO	I
* IN	IT	LOG	ME
* NAT	ON	PLAY	THE
THERE	TO	WE	WILL
WITH	YOU		

THE NUMBER OF WORDS EXCEEDS 2000 AT STORY GROUPK008

THE NUMBER OF WORDS IN THIS STORY GROUP I EXCLUDED [illegible] WORDS

819873

ATTACHMENT 5

Corrective Actions for Certain
Systems Completion Codes

If the user receives a D37** (not enough pages specified), an E22 (I/O requests), or a 322 (time), he may use the following code to continue from where the output was terminated.

Add or change the following cards when indicated:

```
                    .
                    .
                    .
begins in col 6
DCL OLD(2000) CHAR(20);
DCL QFLAG FIXED BIN INIT(0);                    (add)
OLD_GRADE='     ';
                    .
                    .
                    .
                     begins in col 19
                     END;
                     IF QFLAG = 0 THEN GO TO CONTINUE;  (add)
                     PUT SKIP EDIT('GRADE',YR,'STORY NUMBER',
                     STORY)(X(60), A, A(12),
                           .
                           .
                           .
begins in col 10
CONTINUE:OLD_GRADE=GRADE;                           (change)
         IF GRADE = 'xxxx ' THEN QFLAG=1;           (add)*
         NUMB=N;     .
                     .
                     .
```

*xxxx is the story number at which the output should continue printing. e.g., if the program terminated in the middle of story K014, then xxxx should be K014.

Be sure to input the entire deck, even the part that was processed before the program terminated.

**See TN 5-71-44.